UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/797,743	03/09/2004	Yasuyuki Kii	61003 (70820)	3397

21874 7590 06/16/2011
EDWARDS ANGELL PALMER & DODGE LLP
P.O. BOX 55874
BOSTON, MA 02205

EXAMINER	
LIU, ZHENGXI	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
06/16/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/797,743	Applicant(s) KII, YASUYUKI	
	Examiner ZHENGXI LIU	Art Unit 2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 March 2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-11* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-11* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *3/09/2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 3/25/2011 has been entered.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 1, 2, 4, 5, and 9-11 are rejected under 35 U.S.C. 102(b) as being anticipated by Bilodeau et al. (US Patent 6,384,822 B1).

Regarding claim 1, Bilodeau et al. discloses a graphic processing apparatus (*fig. 4*) having a Z-buffer memory storing a Z value representing a depth of a display object when seen from a visual point per pixel (*col. 1, lines 34-39*, stating “when rendering a scene each object in the scene is drawn. A depth value, called the z value, is calculated to indicate the distance to the last polygon drawn for each pixel on the

screen. The z-values for all the pixels on the screen are referred to as the z buffer;" z buffer is a memory that stores z-values; in addition, a scene is always rendered as seen from a visual point) and a pixel memory (*fig. 4, RAM*) storing color data on each pixel for creating an image of a shadowed three-dimensional object having a shadow produced by obstructing a ray of light from a light source by the three-dimensional object (*col. 1, lines 64-67*, stating "the stencil buffer could be set only for pixels in an area in shadow and then the area in the shadow is filled with a transparent gray rectangle or the light for each pixel could be reduced to create a shadow effect;" therefore, the color data on each pixel in a pixel memory is changed by a "transparent gray rectangle" or reduced lighting to create an image with shadow effect; *col. 1, lines 25-28*, disclosing that the objects that create shadows and the shadowed objects are all three-dimensional objects, and stating "for animation, such as utilized in 3-D computer games, shadows must be rendered in real time;" *fig. 2*, showing a three-dimensional object having a shadow produced by obstructing a ray of light from a light source by a three dimensional object; the objects in *fig. 2* are polygons to demonstrate three-dimensional objects in a three-dimensional space for the purpose of illustrating shadow effect in 3-D animation or computer games), comprising:

- a visual-point coordinate conversion processing section (*col. 2, lines 14-15*, disclosing a rendering section, and stating "a technique for using the stencil buffer to render shadows based on a shadow volume 10 is depicted in FIG.2;" rendering scenes is a coordinate conversion process from 3-D to 2-D and based on the viewpoint of a viewer)

- for upon input of graphic data on normal polygons constituting each object including the three-dimensional object and on shadow polygons constituting a shadow volume that defines a shadow space produced by obstructing the ray of light from the light source by the three-dimensional object, converting the graphic data to visual-point coordinates including x-coordinates and y-coordinates and depth values (

    - *col. 2, lines 16-19*, disclosing visual-point coordinate conversion through rendering the normal polygons of the graphic data , and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer;"

    - *col. 2, lines 40-49*, disclosing visual-point coordinate conversion through rendering the shadow polygons of the graphic data, and stating “The invisible front facing polygon 16 is rendered first” and “The invisible back facing polygon 14 is rendered second;" *fig. 2, col. 2, lines 22-29*, disclosing that the front and back facing polygons are shadow polygons;

    - *col. 2, lines 4-8*, disclosing a shadow volume that defines a shadow space produced by obstructing the ray of light from the light source by a three dimensional object, and stating "A

shadow volume for a first scene object is the region of space in which a first object will cast a shadow on any other object appearing in that region. Like any other volume in computer graphics, it is usually represented as a polygon mesh;"

- the normal polygons, front-facing shadow polygons, and back-facing shadow polygons are all rendered (*see explanation above*); the rendering converts the 3-D graphic data of the polygons into an 2D image data based on a viewpoint; the image data is based on visual-point coordinates including x-coordinates and y-coordinates and depth values, because the system discloses the use of two-dimensional rendered pixels [x and y coordinates] and z-values [depth value] to represent the image;

    - *col. 2, line 16-20*, disclosing a visual-point coordinates including x-coordinates and y-coordinates, and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a <u>rectangular grid of pixels</u>, using the z-buffer;" <u>rectangular grid [x and y axes] of pixels are represented by a visual-point coordinates including x-coordinates and y-coordinates</u>;

- *col. 1, lines 49-51*, disclosing a visual-point coordinates include depth values, and stating "the z buffer stores the depth information for each pixel"), and

- outputting the obtained visual-point coordinates and depth values in a state of being sorted into those of front-facing shadow polygons that face front, those of back-facing shadow polygons that face back when seen from the visual point, and those of the normal polygons (

    - *col. 2, lines 21-29*, disclosing that shadow polygons are sorted into front and back facing polygons, and stating "the shadow is marked out in the stencil buffer as follows. The faces of the shadow volume 10 are drawn using invisible polygons 14 and 16. A polygon is front facing if the dot product of its outward normal with the vector from the viewpoint to the scene is negative. … A polygon is back facing if the dot product of its outward normal with the vector from the viewpoint to the scene is positive;"

    - *col. 2, lines 16-19*, disclosing normal/non-shadow polygons are separated and treated differently from shadow polygons, and stating "First, the scene without the shadow is rendered as usual …;"

  - the front and back facing polygons and normal polygons all have been rendered into pixels of a visual-point coordinates, and these pixels also carry depth values; *col. 2, lines 16-19*, disclosing visual-point coordinate conversion of the normal polygons, and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer;" *col. 2, lines 40-49*, disclosing visual-point coordinate conversion of the shadow polygons, and stating "The invisible front facing polygon 16 is rendered first" and "The invisible back facing polygon 14 is rendered second"); and

- a hidden surface removal and shadowing processing section (*col. 1, lines 39-49*, disclosing hidden surface removal through rendering with z-test, and stating "Accordingly, after all the polygons are drawn [rendered] each pixel is left with the value of the front-most surface of the front-most object;" by keeping the front-most surface of the front-most object, hidden surfaces are removed; *col. 2, lines 14-16*, disclosing shadow processing/rendering, and stating "A technique for using the stencil buffer to render shadows based on a shadow volume ...;" the section that removes hidden surfaces and processes shadows is a hidden surface removal and shadowing processing section) for
  - obtaining a coordinate region (*fig. 2, 22*) that is positioned behind the front-facing shadow polygons (*fig. 2, 16*) and in front of the back-facing

shadow polygons (*fig. 2, 14*) when seen from the visual point (*fig. 2, 12*)

- based on the visual-point coordinates, the depth values and the Z-buffer memory after hidden surface removal processing by Z-buffer method is performed on the normal polygons (
    - *col. 2, lines 36-40*, disclosing that the z-test use to locate shadow regions is based on z-buffer memory and depth values, and stating "The z-testing procedure is enabled so that the <u>depth of the shadow volume pixel</u> will be compared with the <u>depth of the scene pixels</u> but z-writes are disabled so that the <u>z-buffer</u> will not be changed by the test;"
    - the z-test used to locate shadow regions is also based on visual-point coordinates, because the test uses rendered shadow volume and scene pixels; rendering converts 3-D graphic data into an 2D image data with a viewpoint and based on x-coordinates and y-coordinates; *col. 2, lines 16-19 and 40-49*, disclosing visual-point coordinate conversion of the normal polygons and shadow polygons through rendering;

- *col. 2, lines 16-19*, disclosing hidden surface removal by Z-buffer is performed on the normal/non-shadow polygons before locating shadow regions, and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer"), and

  - updating color data on pixels in the pixel memory corresponding to the obtained coordinate region to shadow color data (*col. 1, lines 64-67*, stating "the stencil buffer could be set only for pixels in an area in shadow and then the area in the shadow is filled with a transparent gray rectangle or the light for each pixel could be reduced to create a shadow effect;" the area in shadow is the obtained coordinate region, and the color data on pixels in the pixel memory are updated based on transparent gray rectangles or reduced light),

- processing of the back-facing shadow polygons includes obtaining the depth value of each pixel of the back-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is equal to or greater than the corresponding Z value, then the pixel is processed as the back-facing shadow polygon (

  - *col. 1, lines 44-47*, describing a z-test that compares z values of pixels that would be rendered at the same location of a screen, and stating

"The new polygon passes the z-test only if its z [depth] value is <u>less than</u> the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;"

- *col. 2, lines 43-48*, describing the use of z-test on back-facing shadow polygons, and stating "The invisible back facing polygon 14 is rendered second. For each pixel the z-test is conducted and the stencil buffer entry for the pixel is decremented only if the z-value of the back facing pixel passes the standard z-test;" this means: the stencil buffer entry for the pixel will be decremented, only if the depth value of a back-facing polygon is less than a corresponding Z value; <u>therefore, if the depth value is equal to or greater than the corresponding Z value, the pixel will be processed as the back-facing shadow polygon, so that the stencil buffer entry of the pixel is not decremented</u>), and

- processing of the front-facing shadow polygons includes obtaining the depth value of each pixel of the front-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is smaller than the corresponding Z value, then the pixel is processed as the front-facing shadow polygon(
  - *col. 1, lines 44-47*, <u>describing a z-test that compares z values of pixels that would be rendered at the same location of a screen;</u>

- *col. 2, lines 40-45*, describing the use of z-test on front-facing shadow polygons, and stating "The invisible front facing polygon 16 is rendered first. For each pixel the z-test is conducted and the stencil buffer entry for that pixel is incremented only if the z-value of the front facing pixel passes the standard z-test;" this means: only if the depth value of a front-facing shadow polygon is less than a corresponding Z value, the pixel is processed as the front-facing shadow polygon, so that the stencil buffer entry for the pixel is incremented),

- such that the pixels are identified and provided with color representing the shadow if the pixels are associated with a front-facing shadow polygon in front of one of the normal polygons, and a back-facing shadow polygon in back of another of the normal polygons(
  - the Bilodeau et al. invention, illustrated by *fig. 2*, discloses that shadow pixels (e.g., *fig. 2, P2*) are identified through an algorithm associated with a front-facing shadow polygon (*fig. 2, 16*) in front of the normal polygon (*fig. 2, 22*), and a back-facing shadow polygon (*fig. 2, 14*) in back of another of the normal polygons (*fig. 2, 22 or 24*);
  - *fig.2, col. 2, lines 56-63*, describing details of the algorithm, and stating "For pixel P2 representing a second scene polygon 22 located within the shadow volume 10 the pixel fails the z-test when the back facing polygon 14 is drawn, so the stencil entry is not decremented, but

passes the z-test when the front facing polygon 16 is drawn so the stencil buffer is incremented for a net increment of the stencil buffer entry, i.e., the second scene polygon 22 is in the shadow").

Regarding claim 2, Bilodeau et al. further discloses wherein

- the Z-buffer memory and the pixel memory have a capacity for one line in one display screen (
  - *col. 1, lines 34-39*, disclosing the use of a Z-buffer memory, and stating "when rendering a scene each object in the scene is drawn. ... The z-values for <u>all the pixels</u> on the screen are referred to as the <u>z buffer</u>;" *see fig. 4, col. 1, lines 45-48*, disclosing that pixel values are stored in memory, and stating "indicating the present polygon surface is in front of the previous polygon surface <u>stored in the pixel;</u>" the memory that stores pixel data is a pixel memory;
  - the Z-buffer memory and the pixel memory drives a screen area made up of lines of pixels to be displayed on a display screen; *fig. 2*, showing pixels P1, P2, and P3 of the pixel memory that corresponds pixels of to a 2-D screen area in front of the viewpoint (*fig. 2, 12*), and the screen area, as shown in fig. 2, can be viewed as a group of neighboring horizontal/vertical lines; therefore, the pixel memory has a capacity for

one line in one display screen; *col. 1, lines 34-39*, disclosing that Z-buffer memory matches pixel memory pixel by pixel, and stating “A depth value, called the z value, is calculated to indicated the distance to the last polygon drawn for each pixel on the screen;” therefore, Z-buffer also has the capacity for one line in one display screen), and

- the visual-point coordinate conversion processing section and the hidden surface removal and shadowing processing section process per line (
    - disclosing a visual-point coordinate conversion processing section and the hidden surface removal and shadowing processing section process per line (*see analyses provided in claim 1 rejection*);
    - these sections can process per line, because the visual coordinate conversion processing section and the hidden surface removal and shadowing processing section process images on a pixel-by-pixel basis for a rectangular grid of pixels;
        - *col. 2, line 16-20*, disclosing the visual coordinate conversion processing section processes on a pixel by pixel basis, and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer;”

- *col. 1, lines 43-49*, disclosing hidden surface removal section processes on a pixel-by-pixel basis and stating "The new polygon passes the z-test only if its z value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;"

- *col. 1, lines 64-67*, disclosing shadowing processing section processes on a pixel-by-pixel basis, and stating "the stencil buffer could be set only for pixels in an area in shadow and then the area in the shadow is filled with a transparent gray rectangle or the light for each pixel could be reduced to create a shadow effect").

Regarding claim 4, Bilodeau et al. discloses a graphic processing apparatus (*fig. 4*) having a Z-buffer memory storing a Z value representing a depth of a display object when seen from a visual point per pixel (*col. 1, lines 34-39*, stating "when rendering a scene each object in the scene is drawn. A depth value, called the z value, is calculated to indicate the distance to the last polygon drawn for each pixel on the screen. The z-values for all the pixels on the screen are referred to as the z buffer;" z buffer is a memory that stores z-values; in addition, a scene is always rendered as seen from a visual point) and a pixel memory storing color data on each pixel for

creating an image of a shadowed three-dimensional object having shadows produced by obstructing a ray of light from a light source by the three-dimensional object (*col. 1, lines 64-67*, disclosing that color data on each pixel in a pixel memory is changed by a "transparent gray rectangle" or reduced lighting to create shadow effect, and stating "the stencil buffer could be set only for pixels in an area in shadow and then the area in the shadow is filled with a transparent gray rectangle or the light for each pixel could be reduced to create a shadow effect;" *col. 1, lines 25-28*, disclosing that the objects that create shadows and the shadowed objects are all three-dimensional objects, and stating "for animation, such as utilized in 3-D computer games, shadows must be rendered in real time;" *fig. 2*, showing a three-dimensional object having a shadow produced by obstructing a ray of light from a light source by a three dimensional object), comprising:

- a normal polygon conversion section for upon input of graphic data on normal polygons constituting each object including the three-dimensional object, converting the graphic data to visual-point coordinates including x-coordinates and y-coordinates and depth values (*col. 2, lines 16-19*, disclosing normal/non-shadow polygons are converted into visual-point coordinates through rendering, and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer;" therefore, normal polygons are converted into a rectangular grid of pixels through rendering; the rectangular grid [x and y axes] of pixels is represented by a visual-point coordinates including x-

coordinates and y-coordinates; the z-buffer value for each pixel used in rendering carries the depth value; *col. 1, lines 25-28*, disclosing these normal polygons constituting each object including the three-dimensional object, and stating “for animation, such as utilized in 3-D computer games, shadows must be rendered in real time;”);

- a shadow polygon conversion section for upon input of graphic data on shadow polygons constituting a shadow volume that defines a shadow space produced by obstructing the ray of light from the light source by the three-dimensional object (*fig. 2, col. 2, lines 4-8*, disclosing a shadow volume that defines a shadow space produced by obstructing the ray of light from the light source by a three dimensional object, and stating "A shadow volume for a first scene object is the region of space in which a first object will cast a shadow on any other object appearing in that region. Like any other volume in computer graphics, it is usually represented as a polygon mesh"),
    - converting the graphic data to visual-point coordinates including x-coordinates and y-coordinates and depth values (
        - *col. 2, lines 40-49*, disclosing visual-point coordinate conversion through rendering the shadow polygons, and stating “The invisible front facing polygon 16 is rendered first” and “The invisible back facing polygon 14 is rendered second;" *fig. 2, col.*

2, lines 22-29, disclosing that the front and back facing polygons are shadow polygons;

- rendering produces 2D image data based on visual-point coordinates including x-coordinates and y-coordinates and depth values, because the system discloses rendering rectangular grid of pixels and the use of z-values for the pixels;
  - *col. 2, line 16-20*, stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer;" the rectangular grid of pixels [x and y axes] are represented by a visual-point coordinates including x-coordinates and y-coordinates;
  - *col. 1, lines 49-51*, stating "the z buffer stores the depth information for each pixel;" therefore, the visual-point coordinates include depth values for each pixel), and

- outputting the visual-point coordinates and the depth values in a state of being sorted into those of front-facing shadow polygons that face front when seen from a visual point and those of back-facing shadow polygons that face back when seen from the visual point(

  - *col. 2, lines 21-29*, disclosing that shadow polygons are sorted into front and back facing polygons, and stating "the shadow is marked out in the stencil buffer as follows. The faces of the shadow volume 10 are drawn using invisible polygons 14 and 16. A polygon is front facing if the dot product of theist outward normal with the vector from the viewpoint to the scene is negative. ... A polygon is back facing if the dot product of its outward normal with the vector from the viewpoint to the scene is positive;"
  - as the front and back facing polygons have been rendered, the visual-point coordinates and depth values associated with those polygons are obtained and generated as output; *col. 2, lines 40-49*, disclosing visual-point coordinate conversion of the shadow polygons);

- a normal polygon processing section for performing hidden surface removal processing by Z-buffer method on the normal polygons based on the visual-point coordinates and the depth values of the normal polygons and updating color data and a Z value of each pixel in the pixel memory and the Z-buffer memory based on the processing result (
  - *col. 1, lines 39-49*, disclosing hidden surface removal with z-test, and stating "Before the scene is rendered, the z buffer is initialized to a

maximum distance… The new polygon passes the z-test only if its z value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel. Accordingly, after all the polygons are drawn each pixel is left with the value of the front-most surface of the front-most object;" by keeping the front-most surface of the front-most object, hidden surfaces are removed;

- *fig. 2, col. 2, lines 16-19*, disclosing hidden surface removal process is performed on normal/non-shadow polygons through rendering; the removal process is also based on visual-point coordinates, because rendering into 2D images always assumes a viewpoint; the removal process also depends on depth values/z values, because the z-test used in the removal process also uses depth values/z-values of the normal polygons;

- *col. 1, lines 39-49*, disclosing that each pixel is updated to the pixel values of the front most surface after rendering with z-test, and stating "Accordingly, after all the polygons are drawn each pixel is left with the value of the front-most surface of the front-most object;" therefore, each pixel's color data in the pixel memory and Z value in the Z-buffer memory are changed based on the processing result);

- a back-facing shadow polygon processing section for obtaining a coordinate region positioned in front of the back-facing shadow polygons when seen from the visual point based on the visual-point coordinates and the depth values of the back-facing shadow polygons and on the Z values after the hidden surface removal processing is performed (
  - *col. 1, lines 44-47*, describing the z-test as "The new polygon passes the z-test only if its z [depth] value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;"
  - *col. 2, lines 43-48*, disclosing the use of z-test on back-facing shadow polygons, and stating "The invisible back facing polygon 14 is rendered second. For each pixel the z-test is conducted and the stencil buffer entry for the pixel is decremented only if the z-value of the back facing pixel passes the standard z-test;" if the depth value of a back-facing shadow polygon is equal or greater than the corresponding Z value, the stencil buffer entry of the pixel is not decremented, thus identifying a coordinate region positioned in front of the back-facing shadow polygons; this identification process is effective, because the stencil buffer entry of the pixels behind the back-facing shadow polygons will be decremented instead;

  - the z-test used for rendering back-facing shadow polygons is based on the visual-point coordinates, because rendering into a grid of pixels always assumes a viewpoint; z-test also uses depth values/z values;
  - back-facing shadow polygon processing is conducted after the hidden surface removal processing, because the hidden surface removal process of the normal polygons is processed first*; col. 2, lines 16-19*, stating "First, the scene without the shadow is <u>rendered</u> as usual to a bitmap 11, configured as a rectangular grid of pixels, <u>using the z-buffer</u>;"):

- wherein processing of the back-facing shadow polygons includes obtaining the depth value of each pixel of the back-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is equal to or greater than the corresponding Z value, then the pixel is processed as the back-facing shadow polygon (
  - *col. 1, lines 44-47*, <u>describing the z-test</u> as "The new polygon passes the z-test only if its z [depth] value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;"
  - *col. 2, lines 43-48*, describing the use of z-test on back-facing shadow polygons, and stating "The invisible back facing polygon 14 is rendered second. For each pixel the z-test is conducted and the stencil buffer

entry for the pixel is decremented only if the z-value of the back facing pixel passes the standard z-test;" this means that the stencil buffer entry for the pixel will be decremented, only if the depth value of a back-facing polygon is less than a corresponding Z value; <u>therefore, if the depth value of a back-facing polygon is equal to or greater than the corresponding Z value, or when the pixel is positioned in front of the back-facing shadow, the pixel will be processed as the back-facing shadow polygon, so that the stencil buffer entry of the pixel is not decremented</u>);

- a shadow flag memory for storing a flag value representing a visual-point coordinate positioned in front of the back-facing shadow polygons (*col. 2, lines 43-48*, disclosing a stencil buffer as a shadow flag memory for storing a flag value, and stating "The invisible back facing polygon 14 is rendered second. For <u>each pixel the z-test is conducted</u> and the stencil buffer entry for the pixel is decremented only if the z-value of the back facing pixel passes the standard z-test;" if the depth value is equal to or greater than the corresponding Z value, or when a pixel is in front of the back-facing shadow polygon, the pixel is flagged by not changing its value, in contrast to other pixels, the values of which will be decremented); and

- a front-facing shadow polygon processing section for obtaining a coordinate region (*fig. 2, 22*) positioned behind the front-facing shadow polygons (*fig. 2, 16*) and in front of the back-facing shadow polygons (*fig. 2, 14*) when seen

from the visual point (*fig. 2, 12*) based on the visual-point coordinates and the depth values of the front-facing shadow polygons and on the Z values after the hidden surface removal processing is performed and on the flag value, and for updating color data on pixels in the pixel memory corresponding to the obtained coordinate region to shadow color data (

- *col. 1, lines 44-47*, describing the z-test as "The new polygon passes the z-test only if its z [depth] value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;"
- *col. 2, lines 41-45*, disclosing the use of z-test by a processing section on front-facing shadow polygons, and stating "The invisible front facing polygon 16 is rendered first. For each pixel the z-test is conducted and the stencil buffer entry for that pixel is incremented only if the z-value of the front facing pixel passes the standard z-test;" this means: if the depth value of a front-facing polygon is less than a corresponding Z value, or when a pixel is positioned behind the front-facing shadow, the pixel is processed as the front-facing shadow polygon, so that the stencil buffer entry for the pixel is incremented;
- the front-facing shadow polygon processing section with help from the back-facing shadow polygon processing section obtain a coordinate region (*fig. 2, 22*) positioned behind the front-facing shadow polygons

(*fig. 2, 16*) and in front of the back-facing shadow polygons (*fig. 2, 14*) when seen from the visual point (*fig. 2, 12*);

- front-facing shadow polygon processing is conducted based on the visual-point coordinates and the depth values of the front-facing shadow polygons and on the Z values of normal polygons, because the processing renders polygons from 3-D graphic data into a rectangular grid of pixels/visual-point coordinates with x-coordinates and y-coordinates and compares normal polygons' depth values with shadow polygons (*col.1, lines 44-47*) ; *col. 2, lines 16-19*, disclosing visual-point coordinate conversion of the normal polygons of the graphic data , and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer;" *col. 2, lines 41-49*, disclosing visual-point coordinate conversion of the shadow polygons;

- front-facing shadow polygon processing is conducted after the hidden surface removal processing, because the hidden surface removal process of the normal polygons is processed first; *col. 2, lines 16-19*, stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a rectangular grid of pixels, using the z-buffer;"

- *col. 1, lines 64-67*, disclosing the use of flag value, stencil buffer value, to obtain a coordinate region for shadowing, and stating "the stencil buffer could be set only for pixels in an area [coordinate region] in shadow and then the area in the shadow is filled with a transparent gray rectangle or the light for each pixel could be reduced to crate a shadow effect;" also therefore, color data on each pixel in a pixel memory are updated by using "transparent gray rectangle" or reduced lighting to create shadow effect),

- wherein processing of the front-facing shadow polygons includes obtaining the depth value of each pixel of the front-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is smaller than the corresponding Z value, then the pixel is processed as the front-facing shadow polygon (

    - *col. 1, lines 44-47*, describing the z-test as "The new polygon passes the z-test only if its z [depth] value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;"

    - *col. 2, lines 41-45*, disclosing the use of z-test on front-facing shadow polygons, and stating "The invisible front facing polygon 16 is rendered first. For each pixel the z-test is conducted and the stencil buffer entry for that pixel is incremented only if the z-value of the front facing pixel

passes the standard z-test;" this means: if the depth value is less than the corresponding Z value, passing the standard z-test, the pixel is processed as the front-facing shadow polygon, so that the stencil buffer entry for the pixel is incremented),

- such that the pixels are identified and provided with color representing the shadow if the pixels are associated with a front-facing shadow polygon in front of one of the normal polygons, and a back-facing shadow polygon in back of another of the normal polygons(
    - the Bilodeau et al. invention, illustrated by *fig. 2*, discloses that shadow pixels (e.g., *fig. 2, P2*) are identified through an algorithm associated with a front-facing shadow polygon (*fig. 2, 16*) in front of the normal polygon (*fig. 2, 22*), and a back-facing shadow polygon (*fig. 2, 14*) in back of another of the normal polygons (*fig. 2, 22 or 24*);
    - *fig.2, col. 2, lines 56-63*, describing details of the algorithm, and stating "For pixel P2 representing a second scene polygon 22 located within the shadow volume 10 the pixel fails the z-test when the back facing polygon 14 is drawn, so the stencil entry is not decremented, but passes the z-test when the front facing polygon 16 is drawn so the stencil buffer is incremented for a net increment of the stencil buffer entry, i.e., the second scene polygon 22 is in the shadow").

Regarding claim 5, Bilodeau et al. further discloses wherein

- the Z-buffer memory and the pixel memory, and the shadow flag memory have a capacity for one line in one display screen (
    - *col. 1, lines 34-39*, disclosing the use of a Z-buffer memory, and stating "when rendering a scene each object in the scene is drawn. ... The z-values for all the pixels on the screen are referred to as the z buffer;" *see fig. 4, col. 1, lines 45-48*, disclosing that pixel values are stored in memory, and stating "indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;" the memory that stores pixel data is a pixel memory;
    - *col. 1, lines 64-67*, disclosing the use of stencil buffer as the shadow flag memory, and stating "the stencil buffer could be set only for pixels in an area [coordinate region] in shadow and then the area in the shadow is filled with a transparent gray rectangle or the light for each pixel could be reduced to crate a shadow effect;"
    - the Z-buffer memory, the pixel memory, and the shadow flag memory/stencil buffer drive a screen area made up of lines of pixels to be displayed on a display screen; *fig. 2*, showing pixels P1, P2, and P3 of the pixel memory that correspond to pixels of a 2-D screen area in front of the viewpoint (*fig. 2, 12*), and the screen area, as shown in fig. 2, can be viewed as a group of neighboring horizontal/vertical lines;

therefore, the pixel memory has a capacity for one line in one display screen; *col. 1, lines 34-39*, disclosing that Z-buffer memory matches pixel memory pixel by pixel, and stating "A depth value, called the z value, is calculated to indicated the distance to the last polygon drawn for each pixel on the screen;" therefore, Z-buffer also has the capacity for one line in one display screen; *col. 1, lines 52-53*, disclosing that stencil buffer matches pixel memory pixel by pixel, and stating "A stencil buffer is an additional buffer of per-pixel information, <u>much like a z-buffer</u>;" <u>therefore, the stencil buffer also has the capacity for one line in one display screen</u>), and

- the normal polygon conversion section, the shadow polygon conversion section, the normal polygon processing section, the back-facing shadow polygon processing section, and the front-facing shadow polygon processing section process per line(
    - disclosing the normal polygon conversion section, the shadow polygon conversion section, the normal polygon processing section, the back-facing shadow polygon processing section, and the front-facing shadow polygon processing section (*see analyses provided in claim 4 rejection*);

- these sections can process per line through pixel memory, Z-buffer memory, and stencil buffer, because these sections process images on a pixel-by-pixel basis (*see analyses provided in claim 4 rejection*);
  - *col. 2, line 16-20*, disclosing pixel memory and Z-buffer memory operates on a pixel-by-pixel basis, and stating "First, the scene without the shadow is rendered as usual to a bitmap 11, configured as a <u>rectangular grid of pixels</u>, using the z-buffer;"
  - *col. 1, lines 64-67*, disclosing stencil buffer operates on a pixel-by-pixel basis and stating "the stencil buffer could be set only for pixels in an area in shadow and <u>then the area in the shadow is filled with a transparent gray rectangle or the light for each pixel could be reduced to create a shadow effect</u>").

Regarding claim 9, it recites similar limitations as claim 4 but in a method form. The rationale of claim 4 rejection is applied in rejecting claim 9. Furthermore, Bilodeau et al. teaches an algorithm/method, and stating "The standard prior art technique first calculates a shadow volume which is defined by transparent polygons (Crow, F. C., "<u>Shadow Algorithms for Computer Graphics</u>," SIG-GRAPH 77,242-247)."

Regarding claim 10, Bilodeau et al. discloses the graphic processing apparatus as defined in claim 4 running a graphic processing program causing a computer to function as the normal polygon conversion section, the shadow polygon conversion section, the normal polygon processing section, the back-facing shadow polygon processing section, and the front-facing shadow polygon processing section (*col. 2*, showing a pseudo code of an shadow volume algorithm and indicating similar algorithms can be implemented by graphic processing programs).

Regarding Claim 11, it recites similar limitations as claim 10 but in a program storage medium form. The rationale of claim 10 rejection is applied in rejecting claim 11. Furthermore, Bilodeau et al. teaches a program storage medium form (*fig. 4*).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 3 and 6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bilodeau et al. in view of Everitt et al. (Cass Everitt, Mark J. Kilgard. Practical and

Robust Stenciled Shadow Volumes for Hardware-Accelerated Rendering. March 12, 2002. Published on-line at developer.nvidia.com).

Regarding claim 3, Bilodeau et al. discloses the hidden surface removal and shadowing processing section performs processing concerning the shadow polygons (*see claim 1 rejection for detailed analyses*).

However, Bilodeau et al. does not explicitly disclose wherein if a plurality of the shadow volumes are present, the same processing is performed per shadow volume.

Everitt et al. discloses wherein if a plurality of the shadow volumes are present, the same processing is performed per shadow volume (*page 2, right column, lines 21-23*, stating "This [shadow volume algorithm that uses stencil buffer] can be repeated for <u>multiple light sources</u>, clearing the stencil buffer <u>between rendering the shadow volumes</u> and summing the contribution of each light").

At the time of invention, it would have been obvious to a person of ordinary skill in the art to combine Bilodeau et al. with Everitt et al. The suggestion/motivation would have been in order to allow a graphic system to produce graphics for a scene with multiple light sources (*Everitt et al., page 2, right column, lines 21-23*). Real environment often have multiple light sources; computer graphics must manage multiple light sources to produce realistic images. Everitt et al discloses an algorithm similar or identical to the shadow volume algorithm that has been cited by the examiner from Bilodeau et al. (*see Everitt et al., page 2, right column, lines 1-8; Bilodeau et al., col. 2, lines 41-62*). Both are shadow volume multi-pass rendering algorithms using stencil

buffer. Bilodeau et al. does not explicitly state that the algorithm manages multiple shadow volumes and performs processing per shadow volume. The Bilodeau et al. algorithm is probably able to, and Everitt et al. explicitly states so.

Regarding claim 6, Bilodeau et al. further discloses the back-facing shadow polygon processing section and the front-facing shadow polygon processing section perform processing concerning the shadow polygons (*see claim 4 rejection for detailed analyses*).

However, Bilodeau et al. does not explicitly disclose wherein if a plurality of the shadow volumes are present, the same processing is performed per shadow volume.

Everitt et al. discloses wherein if a plurality of the shadow volumes are present, the same processing is performed per shadow volume (*page 2, right column, lines 21-23*, stating "This [shadow volume algorithm using stencil buffer] can be repeated for <u>multiple light sources</u>, clearing the stencil buffer <u>between rendering the shadow volumes</u> and summing the contribution of each light").

The same analysis for combining Bilodeau et al. with Everitt et al. used in the rejection of claim 3 is incorporated herein.

6. Claims 7 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bilodeau et al. in view of Takeuchi (US Patent 6,402,615).

Regarding claim 7, Bilodeau et al. discloses the graphic processing apparatus with the normal polygon conversion section, the shadow polygon conversion section, the normal polygon processing section, the back-facing shadow polygon processing section, and the front-facing shadow polygon processing section as defined in claim 4 (*see claim 4 rejection for detailed analyses*).

However, Bilodeau et al. does not explicitly disclose that the graphics processing and conversion sections of the graphic processing apparatus are included in a portable device.

Takeuchi discloses that the graphics processing and conversion sections of the graphic processing apparatus are included in a portable device (*col. 22, lines 23-25*, stating "Further, it may also be realized using a mobile phone, portable data terminal, car navigation system, or other communications terminal as a platform").

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to incorporate the graphics system disclosed by Bilodeau et al. on a mobile device that receives graphical data over a network as taught by Takeuchi. The motivation for doing so would have been to provide the user with the flexibility to view the graphical data at convenient location while not overburdening the portable device with the storage requirement of the graphical data.

Regarding claim 8, Bilodeau et al. discloses the graphic processing apparatus as defined in claim 7.

However, Bilodeau et al. does not disclose wherein the portable device is connectable to a communication network, and the graphic data is obtained through communications via the communication network.

Takeuchi discloses wherein the portable device is connectable to a communication network, and the graphic data is obtained through communications via the communication network (*col. 5, lines 14-17,* stating “specifically, for example, it is also possible to use the communications interface unit 109 to download the game program from another piece of equipment, not shown, on the network connected through the communications line 111”).

The same analysis for combining Bilodeau et al. with Takeuchi used in the rejection of claim 7 is incorporated herein.

7. Claims 1, 3, 4, 6, and 9-11 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shimizu (U.S. Patent 6,744,430) in view of Bilodeau et al.

Regarding claim 1, Shimizu discloses a graphic processing apparatus (*fig. 10*) having a Z-buffer memory (*fig. 25, 234, Z value buffer*) storing a Z value representing a depth of a display object when seen from a visual point per pixel (*col. 19, lines 17-19*, stating "The Z value buffer 234 stores pixel Z values for each layer prior to the receipt of

the region determination results for each such layer") and a pixel memory (*fig. 22, 70, frame buffer RAM*) storing color data on each pixel (*col. 19, lines 17- 19*, stating "A frame buffer processor 83 consolidates the color data determined by the shading processor 79 into separate frames, subjects those data to treatment (blending), and outputs images for one frame. A dedicated frame buffer RAM 70provides working memory for the frame buffer processor 83 and stores frame data") for creating an image of a shadowed three-dimensional object having a shadow produced by obstructing a ray of light from a light source by the three-dimensional object (*fig. 1*), comprising:

- a visual-point coordinate conversion processing section (pixel data generator 64 described in *lines 51-55 of col. 17*) for upon input of graphic data on normal polygons constituting each object including the three-dimensional object (normal polygon 7 in *fig. 2*; normal polygon c shown in *fig. 26A*) and on shadow polygons constituting a shadow volume (3c and 3b in *fig. 2*; a1 "front surface" of volume ID_0 *line 4 of col. 22*, polygon a2 "back surface" of volume ID_0 in line 4 of col. 22) that defines a shadow space produced by obstructing the ray of light from the light source by the three-dimensional object (*col. 6, lines 10-12*, stating "This light volume 3 is a virtual light space that is produced by the light source 1 and the polygon (object) 2;" *fig. 1, col. 20, lines 24-26*, stating "In FIG. 26A, the triangular column a is described as an example of a shadow volume, and the square column b as an example of a modifier volume"), converting the graphic data to visual-point coordinates including x-coordinates and y-coordinates and depth values (*col. 17, lines 44-*

47, stating "The apex data are configured by screen coordinates (x, y) that indicate positions on the display screen, Z values that indicate depth...;" *col. 11, lines 30-33*, stating "... the pixel data comprising the polygon ID, polygon attribute information, screen coordinates (Sx, Sy), Z values...;" *see also fig. 11, col. 10, lines 47-53*), and

- outputting the obtained visual-point coordinates and depth values (pixel data of *lines 30-33 of col. 11*) in a state of being sorted into those of front- facing shadow polygons that face front (3c in *fig. 2*, and a1 "front surface" of volume ID_0 as described in *line 4 of col. 22*), those of back-facing shadow polygons that face back (3d in *fig. 2*, polygon a2 "back surface" of volume ID_0 as described in *line 4 of col. 22*) when seen from the visual point (*fig. 2, 4, viewpoint*), and those of the normal polygons (*col. 17, lines 59-64*, stating "The sort preprocessor 110 sorts the pixel data sent from the pixel data generator 64, according to Z value, and executes fragment Z buffer processing that extracts the polygon ID closest to the front for each pixel, in each layer 1 to n as viewed from the direction of the view point; " *col. 18, lines 12-14*, stating "The region buffer controllers 120-1 to 120-n determine whether bound layer data input are a volume polygon (shadow volume, modifier volume) or an ordinary polygon... ;" *col. 18, lines 20-28*, stating "The method adopted for updating the region buffers.., based on the results" of volume polygon front/back determinations... "); and

- a hidden surface removal and shadowing processing section (*fig. 23, 110, sort preprocessor; fig. 24, 140, attribute controller, and fig. 22, 83, frame buffer processor*) for
    - o obtaining a coordinate region that is positioned behind the front- facing shadow polygons and in front of the back-facing shadow polygons when seen from the visual point based on the visual-point coordinates (
        - *col. 18, lines 49-51*, stating "The region buffers 220-1 to 220-n store information on whether something is inside or outside a volume (region), pixel by pixel;"
        - Shimizu discloses pixel data is configured by the screen coordinates (*col. 17, lines 52-55*); therefore, the region information recorded in the buffer on a pixel-by-pixel basis defines a coordinate region;
        - if a polygon is inside a volume, then it is behind the front facing shadow polygons and in front of the back facing shadow polygons relative to a viewpoint, and vice versa; *See e.g. fig. 26A*;
        - since a polygon inside a volume is described on a pixel-by-pixel basis, Shimizu discloses obtaining a coordinate region positioned behind the front facing shadow polygons and in front of the back facing shadow polygons; *col. 18, lines 49-51, col.*

22, lines 57-59, disclosing the use of pixels for polygons, and stating "Because this region is inside volume ID_0, the light ID_1 for the relevant pixel(s) is invalidated, based on the volume data type shadow, and output is affected;" *col. 18, lines 10-12*, disclosing differentiation of polygons inside or outside of a shadow volume, and stating "The region buffers 130-1 to 130-n store information (flags) as to whether something is inside or outside a volume (region);"

- *col. 10, line 10*, disclosing front and back facing surfaces with surface normals defining the surface's orientation

), the depth values and the Z- buffer memory after hidden surface removal processing by Z-buffer method is performed on the normal polygons (*col. 21, lines 25-27*, stating "The sort preprocessor (Z buffer) 110 outputs the polygon ID positioned foremost for each pixel, layer by layer"), and

- updating color data on pixels in the pixel memory corresponding to the obtained coordinate region to shadow color data *(col. 17, lines 28-32*, stating "A frame buffer processor 83 consolidates the color data determined by the shading processor 79 into separate frames, subjects those data to treatment (blending), and outputs images for one frame;").

- such that the pixels are identified and provided with color representing the shadow if the pixels are associated with a front-facing shadow polygon in front of one of the normal polygons, and a back-facing shadow polygon in back of another of the normal polygons (
    - *explained previously in this claim rejection* that “a coordinate region that is positioned behind the front- facing shadow polygons and in front of the back-facing shadow polygons;” therefore, this coordinate region of pixels are associated with a front-facing shadow polygon and a back-facing shadow polygon;
    - *explained previously in this claim rejection* that the coordinate region represents normal polygon(s) inside a shadow volume; therefore, the front-facing shadow polygon is in front of one of the normal polygons inside of the shadow volume, and the back-facing shadow polygon is in back of another of the normal polygons inside of the shadow volume;
    - *explained previously in this claim rejection* that the color data of the coordinate region of pixels are updated to show shadow effect).

However, Shimizu does not clearly disclose processing of the back-facing shadow polygons includes obtaining the depth value of each pixel of the back-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is equal to or greater than the corresponding Z value, then the pixel is processed as the back-facing shadow polygon,

and processing of the front-facing shadow polygons includes obtaining the depth value of each pixel of the front-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is smaller than the corresponding Z value, then the pixel is processed as the front-facing shadow polygon.

These limitations, in short, are the detailed techniques used to determine if a pixel of a normal polygon is inside of a shadow volume: in front of back-facing shadow polygons and behind front-facing shadow polygons.

Bilodeau et al. discloses these detailed techniques used to determine if a pixel of a normal polygon is inside of a shadow volume, specifically the steps of

- processing of the back-facing shadow polygons includes obtaining the depth value of each pixel of the back-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is equal to or greater than the corresponding Z value, then the pixel is processed as the back-facing shadow polygon(
    - *col. 1, lines 44-47*, <u>describing a z-test that compares z values of pixels that would be rendered at the same location of a screen,</u> and stating “The new polygon passes the z-test only if its z [depth] value is <u>less than</u> the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;”
    - *col. 2, lines 43-48*, describing the use of z-test on back-facing shadow polygons, and stating “The invisible back facing polygon 14 is rendered

second. For each pixel the z-test is conducted and the stencil buffer entry for the pixel is decremented only if the z-value of the back facing pixel passes the standard z-test;" this means: the stencil buffer entry for the pixel will be decremented, only if the depth value of a back-facing polygon is less than a corresponding Z value; <u>therefore, if the depth value is equal to or greater than the corresponding Z value, the pixel will be processed as the back-facing shadow polygon, so that the stencil buffer entry of the pixel is not decremented</u>), and

- processing of the front-facing shadow polygons includes obtaining the depth value of each pixel of the front-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is smaller than the corresponding Z value, then the pixel is processed as the front-facing shadow polygon(
    - *col. 1, lines 44-47*, <u>describing a z-test that compares z values of pixels that would be rendered at the same location of a screen;</u>
    - *col. 2, lines 40-45*, <u>describing the use of z-test on front-facing shadow polygons</u>, and stating "The invisible front facing polygon 16 is rendered first. For each pixel the z-test is conducted and the stencil buffer entry for that pixel is incremented only if the z-value of the front facing pixel passes the standard z-test;" this means: <u>only if the depth value of a front-facing shadow polygon is less than a corresponding Z value, the pixel is</u>

processed as the front-facing shadow polygon, so that the stencil buffer entry for the pixel is incremented).

At the time of invention, it would have been obvious to a person of ordinary skill in the art to combine Shimizu with Bilodeau et al. The suggestion/motivation would have been in order to compare z-values of shadow polygons with normal polygons to determine if the normal polygons are inside of a shadow volume. Shimizu discloses sorting/comparing polygons based on depth/z-values, and stating "It is assumed here that the polygon ID1 is positioned in the foreground, and that the polygon ID2 is positioned in back. At pixel b where the polygon ID1 and polygon ID2 overlap, pixel data for the polygon ID1 and pixel data for the polygon ID2 are stored, in Z value order" (*col. 11, lines 41-47*). Shimizu also discloses determining whether a normal polygon is inside a shadow volume or not, and stating "The region buffers 220-1 to 220-n store information on whether something is inside or outside a volume (region), pixel by pixel" (*col. 18, lines 49-51*). Therefore, Shimizu clearly demonstrates its need and capability of using methods similar to Bilodeau et al.'s z-test.

Regarding claim 3, Shimizu further discloses if a plurality of the shadow volumes are present, the hidden surface removal and shadowing processing section performs processing concerning the shadow polygons per shadow volume (*col. 18, lines 15-17*, stating "When a volume polygon (shadow volume modifier volume) has been input, the region buffer controllers 120-1 to 120-n update the region buffers;" *col. 18, lines 20-23*, stating "The method adopted for updating the region buffers may be a method

wherewith in and out are inverted, in pixel units, every time a volume polygon is input... ;" see also attribute modulator B: *lines 53-55 of col. 18 and lines 59-61 of col. 18*).

Regarding claim 4, Shimizu discloses a graphic processing apparatus (*fig. 10*) having a Z-buffer memory (*fig. 25, 234, Z value buffer*) storing a Z value representing a depth of a display object when seen from a visual point per pixel (*col. 19, lines 17-19*, stating "The Z value buffer 234 stores pixel Z values for each layer prior to the receipt of the region determination results for each such layer") and a pixel memory (*fig. 22, 70, frame buffer RAM*) storing color data on each pixel (*col. 19, lines 17-19*, stating "A frame buffer processor 83 consolidates the color data determined by the shading processor 79 into separate frames, subjects those data to treatment (blending), and outputs images for one frame. A dedicated frame buffer RAM 70 provides working memory for the frame buffer processor 83 and stores frame data") for creating an image of a shadowed three-dimensional object having shadows produced by obstructing a ray of light from a light source (*fig. 1, 1*) by the three- dimensional object (*fig. 1, 2*), comprising:

- a normal polygon conversion section (pixel data generator 64 described in lines *51-55 of col. 17*) for upon input of graphic data on normal polygons constituting each object including the three-dimensional object, converting the graphic data to visual-point coordinates including x-coordinates and y-coordinates and depth values (*col. 17, lines 44-47*, stating "The apex data are configured by screen coordinates (x, y) that indicate positions on the display screen, Z values that indicate depth... ;" *col. 11, lines 30-33*, stating "... the

pixel data comprising the polygon ID, polygon attribute information, screen coordinates (Sx, Sy), Z values... "; *see also fig. 11, col. 10, lines 47-53*);

- a shadow polygon conversion section (pixel data generator 64 described in *lines 51-55 of col. 17*) for upon input of graphic data on shadow polygons constituting a shadow volume that defines a shadow space produced by obstructing the ray of light from the light source by the three-dimensional object (*fig. 1, 3*),
  - converting the graphic data to visual-point coordinates including x-coordinates and y-coordinates and depth values (*col. 17, lines 44-47*, stating "The apex data are configured by screen coordinates that indicate positions on the display screen, Z values that indicate depth... ;" *col. 11, lines 30-33*; *fig. 11*), and
  - outputting the obtained visual-point coordinates and depth values (pixel data of *lines 30-33 of col. 11*) in a state of being sorted into those of front-facing shadow polygons that face front (3c in *fig. 2*," a1 "front surface" of volume ID_0 in *line 4 of col. 22*), those of back-facing shadow polygons that face back (3d in *fig. 2*, polygon a2 "back surface" of volume ID_0 in *line 4 of col. 22*) when seen from the visual point (*fig. 2, 4, view point*; sort processing described in *lines 59-64 of col. 17*; *col. 18, lines 12-14; col. 18, lines 20-28*);
- a normal polygon processing section (*fig. 23, 110, sort preprocessor; fig. 24, 140, attribute controller, and fig. 22, 83, frame buffer processor*) for

performing hidden surface removal processing by Z-buffer method on the normal polygons based on the visual-point coordinates and the depth values of the normal polygons (*col. 21, lines 25-27*, stating "The sort preprocessor (Z buffer) 110 outputs the polygon ID positioned foremost for each pixel, layer by layer") and updating color data and a Z value of each pixel in the pixel memory and the Z-buffer memory based on the processing result (*col. 21, lines 25-27*, stating "The sort preprocessor (Z buffer) 110 outputs the polygon ID positioned foremost for each pixel, layer by layer;" *col. 17, lines 28-32*, stating "A frame buffer processor 83 consolidates the color data determined by the shading processor 79 into separate frames, subjects those data to treatment (blending), and outputs images for one frame");

- a back-facing shadow polygon processing section (*fig. 23, 110, sort preprocessor; fig. 24, 140, attribute controller, and fig. 22, 83, frame buffer processor*) for obtaining a coordinate region positioned in front of the back-facing shadow polygons (*col. 20, lines 9-11*, stating "The triangular column a is defined by five polygons, namely by a front surface a1, back surface a2...;" *col. 20, lines 25-27*, stating "In FIG. 26A, the triangular column a is described as an example of a shadow volume...;" the operations of the region buffers for determining a region for back surface a2 are illustrated in *figs. 29, 30, 31, 34, 37*) when seen from the visual point based on the visual-point coordinates (*col. 18, lines 49-51*, stating "The region buffers 220-1 to 220-n store information on whether something is inside or outside a volume (region), pixel

by pixel;" *col. 18, lines 10-12*) and the depth values of the back-facing shadow polygons and on the Z values after the hidden surface removal processing is performed (*col. 21, lines 22-29*, stating "Next, with a delineation start instruction... The sort preprocessor (Z buffer) 110 outputs the polygon ID positioned foremost for each pixel, layer by layer");

- a shadow flag memory (*region buffers 130-1 to 130-n*) for storing a flag value representing a visual-point coordinate positioned in front of the back-facing shadow polygons (*col. 18, lines 10-12*, stating "The region buffers 130-1 to 130-n store information (flags) as to whether something is inside or outside a volume (region)."); and
- a front-facing shadow polygon processing section (*fig. 23, 110, sort preprocessor; fig. 24, 140, attribute controller, and fig. 22, 83, frame buffer processor*) for obtaining a coordinate region positioned behind the front-facing shadow polygons (*col. 20, lines 9-11*, stating "The triangular column a is defined by five polygons, namely by a front surface a1, back surface a2... ;" *col. 20, lines 25-27*, stating "In FIG. 26A, the triangular column a is described as an example of a shadow volume... ;" the operations of the region buffers for determining a coordinate region for front surface a1 are illustrated in *figs. 29, 30, 31, 34, 37*) and in front of the back-facing shadow polygons when seen from the visual point based on the visual-point coordinates (*col. 18, lines 49-51*) and the depth values of the front-facing shadow polygons and on the Z values after the hidden surface removal processing is performed (*col. 21,*

lines 22-29) and on the flag value and for updating color data on pixels in the pixel memory corresponding to the obtained coordinate region to shadow color data (*fig. 44, col. 22, lines 57-59*, stating "Because this region is inside volume ID_0, the light ID_1 for the relevant pixel(s) is invalidated, based on the volume data type shadow, and output is affected;" *col. 23, lines 11-15*, stating "The pixel data for the polygon c described earlier pass through a layer controller 77 and attribute modulator 78, and a pixel delineation such as diagrammed in FIG. 45 is made by the shading processor 79, texture processor 80, and frame processor 83"),

- such that the pixels are identified and provided with color representing the shadow if the pixels are associated with a front-facing shadow polygon in front of one of the normal polygons, and a back-facing shadow polygon in back of another of the normal polygons (
    - *explained previously in this claim rejection* that "a coordinate region that is positioned behind the front- facing shadow polygons and in front of the back-facing shadow polygons;" therefore, this coordinate region of pixels are associated with a front-facing shadow polygon and a back-facing shadow polygon;
    - *explained previously in this claim rejection* that the coordinate region represents normal polygon(s) inside a shadow volume; therefore, the front-facing shadow polygon is in front of one of the normal polygons

inside of the shadow volume, and the back-facing shadow polygon is in back of another of the normal polygons inside of the shadow volume;

- *explained previously in this claim rejection* that the color data of the coordinate region of pixels are updated to show shadow effect).

Applicant's invention and Shimizu use different terminologies for the same/similar concepts. Clarification and emphasis are given to the following:

- Shimizu discloses pixel data is configured by the screen coordinates (*col. 17, lines 52-55*); therefore, the region information recorded in the buffer on a pixel-by-pixel basis defines a coordinate region;
- if a polygon is inside a volume, then it is behind the front facing shadow polygons and in front of the back facing shadow polygons relative to a viewpoint, and vice versa; *See e.g. fig. 26A*; since a polygon inside a volume is described on a pixel-by-pixel basis, Shimizu discloses obtaining a coordinate region positioned behind the front facing shadow polygons and in front of the back facing shadow polygons; *col. 18, lines 49-51, col. 22, lines 57-59*, disclosing the use of pixels for polygons, and stating "Because this region is inside volume ID_0, the light ID_1 for the relevant pixel(s) is invalidated, based on the volume data type shadow, and output is affected;" *col. 18, lines 10-12*, disclosing differentiation of polygons inside or outside of a shadow volume, and stating "The region buffers 130-1 to 130-n store information (flags) as to whether something is inside or outside a volume

(region);" *col. 10, line 10*, disclosing front and back facing surfaces with surface normals defining the surface's orientation.

However, Shimizu does not clearly disclose wherein processing of the back-facing shadow polygons includes obtaining the depth value of each pixel of the back-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is equal to or greater than the corresponding Z value, then the pixel is processed as the back-facing shadow polygon, and wherein processing of the front-facing shadow polygons includes obtaining the depth value of each pixel of the front-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is smaller than the corresponding Z value, then the pixel is processed as the front-facing shadow polygon.

These limitations, in short, are the detailed techniques used to determine if a pixel of a normal polygon is inside of a shadow volume: in front of back-facing shadow polygons and behind front-facing shadow polygons.

Bilodeau et al. discloses these detailed techniques used to determine is a pixel of a normal polygon is inside of a shadow volume, specifically

- wherein processing of the back-facing shadow polygons includes obtaining the depth value of each pixel of the back-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is equal to or greater than the corresponding Z value, then the pixel is processed as the back-facing shadow polygon (

- *col. 1, lines 44-47*, describing the z-test as "The new polygon passes the z-test only if its z [depth] value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;"
- *col. 2, lines 43-48*, describing the use of z-test on back-facing shadow polygons, and stating "The invisible back facing polygon 14 is rendered second. For each pixel the z-test is conducted and the stencil buffer entry for the pixel is decremented only if the z-value of the back facing pixel passes the standard z-test;" this means that the stencil buffer entry for the pixel will be decremented, only if the depth value of a back-facing polygon is less than a corresponding Z value; therefore, if the depth value of a back-facing polygon is equal to or greater than the corresponding Z value, or when the pixel is positioned in front of the back-facing shadow, the pixel will be processed as the back-facing shadow polygon, so that the stencil buffer entry of the pixel is not decremented);

- wherein processing of the front-facing shadow polygons includes obtaining the depth value of each pixel of the front-facing shadow polygons, comparing the depth value with a corresponding Z value obtained from the Z-buffer memory, and if the depth value is smaller than the corresponding Z value, then the pixel is processed as the front-facing shadow polygon (

- *col. 1, lines 44-47*, describing the z-test as “The new polygon passes the z-test only if its z [depth] value is less than the z value stored in the z buffer, indicating the present polygon surface is in front of the previous polygon surface stored in the pixel;”
- *col. 2, lines 41-45*, disclosing the use of z-test on front-facing shadow polygons, and stating “The invisible front facing polygon 16 is rendered first. For each pixel the z-test is conducted and the stencil buffer entry for that pixel is incremented only if the z-value of the front facing pixel passes the standard z-test;" this means: if the depth value is less than the corresponding Z value, passing the standard z-test, the pixel is processed as the front-facing shadow polygon, so that the stencil buffer entry for the pixel is incremented),

The same analysis for combining Shimizu with Bilodeau et al. used in the rejection of claim 1 is incorporated herein.

Regarding claim 6, Shimizu further discloses if a plurality of the shadow volumes are present, the back-facing shadow polygon processing section and the front facing shadow polygon processing section perform processing concerning the shadow polygons per shadow volume (*col. 18, lines 15-17*, stating "When a volume polygon (shadow volume, modifier volume) has been input, the region buffer controllers 120-1 to 120-n update the region buffers;" *col. 18, lines 20-23*, stating "The method adopted for updating the region buffers may be a method wherewith in and out are inverted, in pixel

units, every time a volume polygon is input...;" see also attribute modulator B in *lines 53-55 of col. 18 and lines 59-61 of col. 18*).

Regarding Claim 9, it recites limitations similar in scope to those presented in claim 4 as a method. Shimizu invention is embodied in a method as shown in *line 10 of col. 2*. The limitations of claim 9 are rejected with the rationale presented to reject the corresponding elements in the apparatus disclosed in claim 4.

Regarding claim 10, Shimizu discloses the graphic processing apparatus as defined in claim 4 running a graphic processing program causing a computer to function as the normal polygon conversion section, the shadow polygon conversion section, the normal polygon processing section, the back-facing shadow polygon processing section, and the front-facing shadow polygon processing section (*col. 15, lines 49-55*, stating "In the example diagrammed here in FIG. 20, however, a general purpose computer is" used and image processing (geometry processing, pixel data generation processing, pixel sorter processing, attribute alteration processing, and rendering processing) is implemented by a software program or programs").

Regarding claim 11, Shimizu discloses a program storage medium allowing computer to read, characterized in that the graphic processing program as defined in claim 10 is stored (*col. 16, lines 7-12*, stating "The recording media for providing the

computer programs for executing the processing described in the foregoing include, in addition to such information recording media as magnetic disks and CD-ROMs... ").

8. Claims 2 and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shimizu in view of Bilodeau et al. and Kelley et al. (U.S. Patent 5,517,603).

Regarding claims 2 and 5, Shimizu in view of Bilodeau et al. discloses the limitations of parent claims 1 and 4, respectively, as well as "the Z-buffer memory and the pixel memory and shadow flag memory have a capacity for one line in one display screen" (*col. 13, lines 51-53*, stating "When it is determined in step S2 that all of the polygon pixel data for one frame have been stored in the sort buffer 66, step 3 is advanced to... "). One of ordinary skill in the art would recognize that if the memory has the capacity to store all of the data for one frame then clearly the memory has the capacity for one line of that frame. With regard to claim 2, Shimizu in view of Bilodeau et al. does not expressly disclose "the visual-point coordinate conversion processing section and the hidden surface removal and shadow processing section process per line." With regard to claim 5, Shimizu in view of Bilodeau et al. does not disclose "the normal polygon conversion section, the shadow polygon conversion section, the normal polygon processing section, the back-facing shadow polygon processing section, and the front-facing shadow polygon section process per line."

With regard to claims 2 and 5, Kelley et al discloses "the Z-buffer memory, the pixel memory, and the shadow flag memory have a capacity for one line in one display

screen (*col. 32, lines 24-26*, stating "FIG. 12 is a functional block diagram of a stage 2/3 processing unit. A RAM 1201 and a RAM 1202 comprise the dual buffers and consist of one scanline of memory each;" *col. 31, lines 63-66*, stating "When performing scanline Z-buffering or operating as a compositing engine, both require at least one complete scanline of memory"), and "the visual- point coordinate conversion processing section and the hidden surface removing (*col. 15, lines 17-20*) and shadowing processing (*col. 21, lines 47-50; col. 22, lines 4-7*) section processes per line," as recited in claim 2, and "the normal polygon conversion section (*col. 14, lines 37- 40; col. 14, lines 45-47*), the shadow polygon conversion section (*col. 21, lines 39-45; col. 14, lines 45-47; col. 24, lines 18-19*), the normal polygon processing section (col. 15, lines 17-20; *col. 15, lines 45-47*), the back-facing shadow polygon processing section (*line 65 of col. 21 through line 2 of col. 22*), and the front- facing shadow polygon processing section (*col. 21, lines 54-59; col. 22, lines 2-7*) process per line," as recited in claim 5 (*col. 3, lines 66-67*, stating "In the scanline approach the 3- D image is rendered a scanline at a time, rather than an object at a time;" *col. 6, lines 10-13*, stating "Utilizing a scanline approach for rendering a 3-D graphical image, alternative rendering device configurations provide scalable rendering performance. "). While Kelley et al does not use the language "one line of shadow flag memory," one of ordinary skill in the art would recognize the system operates by performing the operations one scanline at a time, and computes a shadow count for each pixel in each scanline from the statement *lines 1-4 of col. 22*: "A volume entirely in front of the pixel will generate one increment and one decrement at that pixel, leaving the shadow count unchanged."

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to perform the operations disclosed by Shimizu in view of Bilodeau et al. per line as taught by Kelley et al. The motivation for doing so would have been to provide the system with the flexibility to process the pixels out of order or in parallel. Kelley et al discloses the advantages of scanline independence for "Parallel Rendering Pipelines" (*col. 37, lines 5-20*). Therefore, it would have been obvious to modify Shimizu in view of Bilodeau et al. with the teachings of Kelley et al to obtain the invention specified in claims 2 and 5.

9. Claims 7 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Shimizu in view of Bilodeau et al. and Takeuchi.

With regard to claim 7, Shimizu in view of Bilodeau et al. shows the limitations of parent claim 4, but does not show "a portable device." Takeuchi discloses a graphics system on "a portable device" (*col. 22, lines 23- 25*, stating "Further, it may also be realized using a mobile phone, portable data terminal, car navigation system, or other communications terminal as a platform").

With regard to claim 8, Shimizu in view of Bilodeau et al. shows the limitations of claim 4 on which claim 8 depends, but does not show "a communication network." Takeuchi discloses "the portable device is connectable to a communication network, and the graphic data is obtained through communications via the communication network" (*col. 5, lines 14-17*, stating "Specifically, for example, it is" also possible to use

the communications interface unit 109 to download the game program from another piece of equipment, not shown, on the network connected through the communications line 111").

At the time of the invention, it would have been obvious to a person of ordinary skill in the art to incorporate the graphics system disclosed by Shimizu in view of Bilodeau et al. on a mobile device that receives graphical data over a network as taught by Takeuchi. The motivation for doing so would have been to provide the user with the flexibility to view the graphical data at convenient location while not overburdening the portable device with the storage requirement of the graphical data. Therefore, it would have been obvious to combine Shimizu in view of Bilodeau et al. with Takeuchi to obtain the invention specified in claims 7 and 8.

Response to Arguments

10. Applicant's arguments with respect to claims 1, 4, and 9 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- Franklin C. Crow. 1977. Shadow algorithms for computer graphics. SIGGRAPH Comput. Graph. 11, 2 (July 1977), 242-248. This is a seminal paper on shadow volume algorithms.

- US Patent 5,043,922 by Matsumoto on a graphics system that generates shadows by using a depth buffer.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ZHENGXI LIU whose telephone number is (571)270-7509. The examiner can normally be reached on Monday - Friday (8 a.m. - 4:30 p.m.).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kee Tung can be reached on (571)272-7794. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/KEE M TUNG/
Supervisory Patent Examiner, Art Unit 2628
/ZHENGXI LIU/
Examiner, Art Unit 2628